

IN THE CLAIMS:

Please amend the claims as shown below.

1. (Currently amended) A method to map a storage environment data object, comprising;
- receiving a reference to the data object in a first storage environment, wherein the data object resides in a second storage environment;
- generating a first data structure from the reference representing one or more physical locations of the data object within the second storage environment;
- retaining the first data structure in the first storage environment to interface with one or more of the physical locations of the data object from the first storage environment; and
- updating the first data structure when changes occur to the data object in the second storage environment.
2. (Original) The method of claim 1, wherein in retaining the first data structure one or more additional references access the data object using the first data structure.
3. (Original) The method of claim 1, wherein receiving the reference an operating system of the first storage environment does not support the second storage environment.
4. (Original) The method of claim 1, wherein during generation one or more extents of the data object within the second storage environment are provided.
5. (Original) The method of claim 4, wherein the generation further includes detecting a mirroring of the data object on at least two storage devices within the second storage environment.

6. (Original) The method of claim 1, wherein during generation metadata associated with the second storage environment and the data object are provided.
7. (Original) The method of claim 1, wherein in retaining the first data structure the first data structure is validated with one or more subsequent references made to access the data object.
8. (Original) The method of claim 1, wherein the method is used to interface a first database using the first storage environment with a second database using the second storage environment.
9. (Currently amended) A method to represent a data storage object, comprising:
identifying one or more storage locations for the data storage object housed within a first storage environment;
assembling a hierarchical map representing a path to one or more of the storage locations;
using the map in a second storage environment to access the data storage object;
updating the map when changes occur to the data storage object in the first storage environment.
10. (Original) The method of claim 9, wherein while assembling the map attribute data are acquired and associated with the first storage environment.
11. (Original) The method of claim 10, wherein assembling further includes acquiring attribute data associated with the data storage object.
12. (Original) The method of claim 9, wherein in identifying one or more of the storage locations, the data storage object is identified as at least one of a file system, a file, a database, a volume, and a portion of data within a file.

13. (Original) The method of claim 9, wherein the method is repetitively processed for one or more additional data objects residing in the first storage environment.
14. (Original) The method of claim 13, wherein the method is used to create an image or copy of the first storage environment in the second storage environment.
15. (Previously presented) A first computer readable medium having a data map representing a data object residing on a second computer readable medium, the map comprising:
- a first node representing the data object;
 - a file system node representing a file system on the second computer readable medium;
 - a volume node representing a volume manager associated with the file system;
 - one or more partition nodes managed by the volume manager;
 - one or more disk identifications representing one or more storage devices housing the data object; and
- wherein the map is update when changes are detected in any of the nodes.
16. (Original) The map of claim 15, wherein the map is represented as a tree data structure on the computer readable media.
17. (Original) The map of claim 15, wherein each node includes metadata.
18. (Original) The map of claim 15, wherein the map is used by an accessing set of executable instructions having access to a second file system which is incompatible with the first file system.
19. (Original) The map of claim 18, wherein the data object is referenced and modified by the accessing set of executable instructions from the second file system.

20. (Original) The map of claim 19, wherein the map is updated if one or more values associated with one or more of the nodes or identifications are modified.

21. (Previously presented) A storage environment system, comprising:
a first file system;
a second file system;
one or more data objects residing in the second file system and capable of being referenced within the first file system;
a map generated to represent one or more physical locations for each of the one or more data objects and used by the first file system when one or more of the data objects are referenced; and
wherein the map is updated when changes are detected and associated with the one or more data objects.

22. (Original) The system of claim 21, wherein the map is modified when one or more of the physical locations for each of the one or more data objects changes.

23. (Original) The system of claim 21, wherein the file systems reside in different computing environments.

24. (Original) The system of claim 23, wherein the file systems operate within different operating systems.

25. (Original) The system of claim 24, wherein the map is used to replicate the second file system within the first file system in a first file system format.